# Zadanie 1 Modelowanie dyfuzji za pomocą metody kratowego równania Boltzmanna (Lattice Boltzmann Method - LBM)

## Piotr Mirosław

## Inżynieria Obliczeniowa 5 semestr grupa 2

1. Przygotowanie:

Do wykonania zadania użyto języka Python; wysokopoziomowego języka programowania, który jest zorientowany obiektowo, posiada wiele wbudowanych bibliotek, a pisanie w tym języku jest dla mnie najwygodniejsze.

Środowiskiem programowania jest PyCharm, który poprawia składnię, podpowiada nazwy funkcji, pozwala na łatwe instalowanie dodatkowych pakietów.

Do wizualizacji zastosowano bibliotekę Pillow, która pozwala interpreterowi języka Python na przetwarzanie obrazów.

1. Program składa się z 4 części:

- Moduł przygotowujący automat komórkowy.

- Algorytm streamingu

- Algorytm kolizji

- Główna pętla programu

Na początku zaimportowano potrzebne biblioteki oraz zdefiniowano następujące stałe:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Random – biblioteka implementująca funkcje losowe

PIL – biblioteka do przetwarzania obrazu

Frames – zmienna przechowująca kolejne „klatki”, by na końcu wygenerować wizualizację działania automatu.

ITERATIONS – liczba iteracji, tu równa 500

SIZE – rozmiar automatu komórkowego – tu równy 100

CONDITION – warunek pod którym generowane są pełne komórki. Gdy równy 0, wtedy najbardziej prawdopodobne, że komórki będą generowane z jednakowym prawdopodobieństwem we wszystkich kierunkach, gdy równy np. 300 wtedy jeden kierunek jest najbardziej uprzywilejowany i to w nim najczęściej pojawiać się będą pełne komórki.

DIFFCOEF = współczynnik tutaj równy 0.1, w celu przyspieszenia wizualizacji.

Zdefiniowano również pomocniczą funkcję condensation, która oblicza stężenie gazu w komórce.

### Automat komórkowy - wypełnianie

Obraz zawierający tekst

Opis wygenerowany automatycznie

Automat komórkowy jest trój wymiarową tablicą o rozmiarze SIZExSIZEx4, czyli jest to kwadrat o boku = SIZE, trzeci wymiar równy 4 to kierunki wpływu i wypływu cząstek gazu do komórki. Na początku automat wypełniony zostaje zerami. Następnie w pętli brzegi, oraz przegroda przyjmują wartość 2 – kolor biały na wizualizacji. W lewej części pojawiają się cząstki. Zatem lewa część automatu jest wypełniona, podczas gdy prawa jest pusta.

### Algorytm streamingu

Obraz zawierający tekst

Opis wygenerowany automatycznie

Podobnie jak w poprzednim zadaniu, cząstki (a w zasadzie strumień gazu) przemieszcza się w zadanym kierunku. Przy czym z komórki wychodzi pewna funkcja składowa a nie cząstka, funkcje te obliczane są w algorytmie kolizji:

### Algorytm kolizji

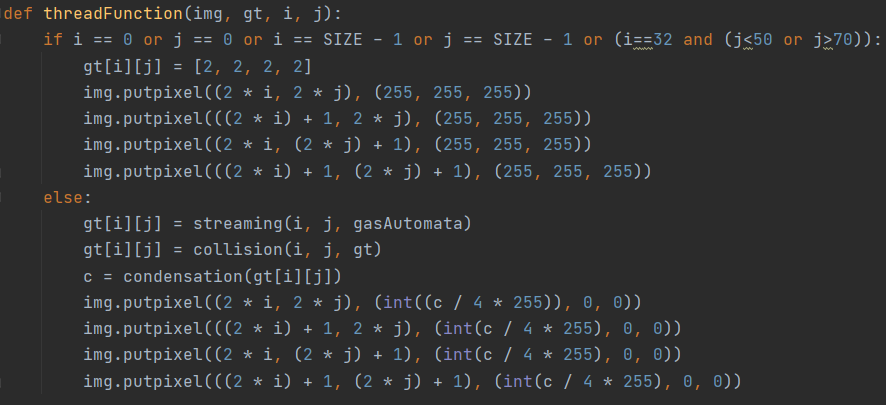
Obraz zawierający tekst

Opis wygenerowany automatycznie

W komórce w danej chwili czasu znajduje się jakaś porcja gazu, która przykładowo = 4, gdy wszystkie 4 elementy komórki są równe 1. W funkcji kolizji obliczane jest stężenie gazu w komórce, a wszystkie wyjścia cząsteczki są ustawiane zgodnie ze wzorem:



### Pętla główna programu:



Obraz zawierający tekst, zrzut ekranu, monitor, zamknąć

Opis wygenerowany automatycznie

Pętla wykonuje się tyle razy ile jest iteracji. W każdym kolejnym kroku tworzony jest nowy pusty automat gasTemp, oraz jego wizualizacja – image\_temp. Następnie w podwójnej zagnieżdżonej pętli przechodzi się po wszystkich komórkach automatu i dokonuje się na nich operacji streamingu i kolizji, które wywoływane są w funkcji threadFunction (nazwa stworzona tak, by w przyszłości można było program zrównoleglić). Chyba, że komórka jest ścianą lub przegrodą, wtedy z komórką nie dzieje się nic, tylko obszar przez nią zajmowany jest kolorowany na biało.

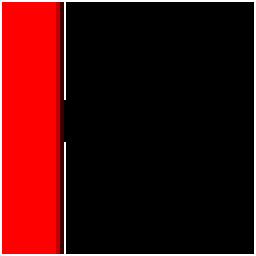
Stężenie w komórce jest przedstawione jako poziom czerwieni. Im jaśniejsza czerwień, tym większe stężenie gazu w danej komórce. Czarna komórka oznacza, że nie ma w niej gazu.

Z wizualizacji można odczytać, że początkowo gaz znajdujący się w lewej komorze, poprzez szczelinę w przegrodzie dostaje się do prawej komory, tak, by na końcu wypełnić cały dostępny obszar i pokolorować równomiernie wszystkie komórki – jednakowe stężenie w całej objętości.

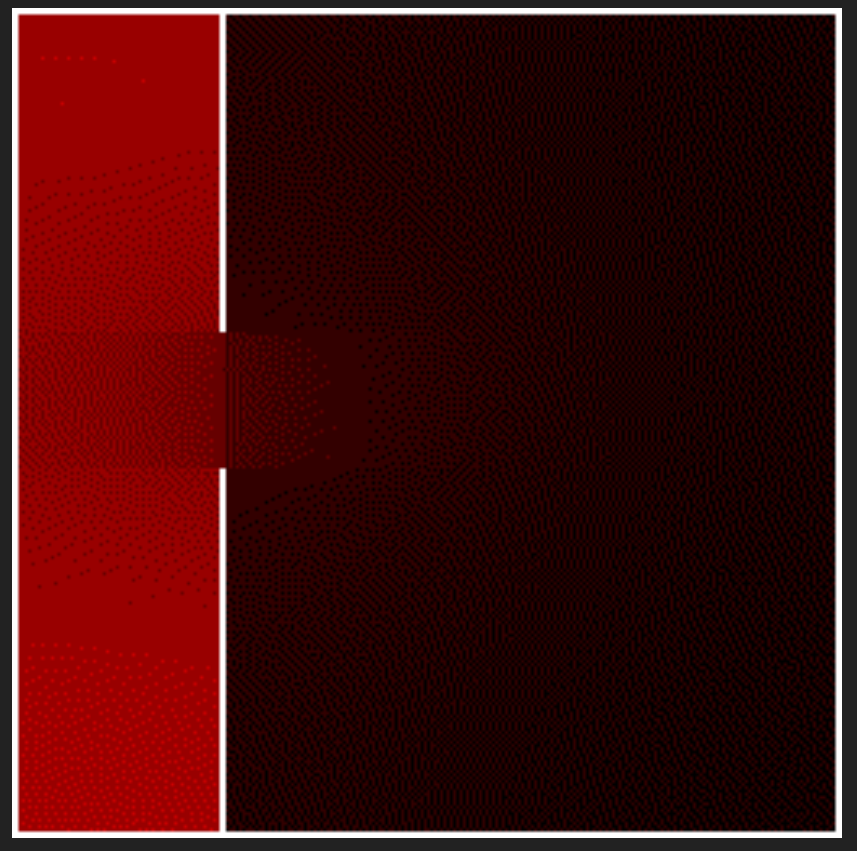
Po przeiterowaniu po wszystkich elementach wygenerowany obraz jest dołączany do listy frames, by po zakończeniu pętli mógł być zwizualizowany w pliku gif.

# Wyniki:

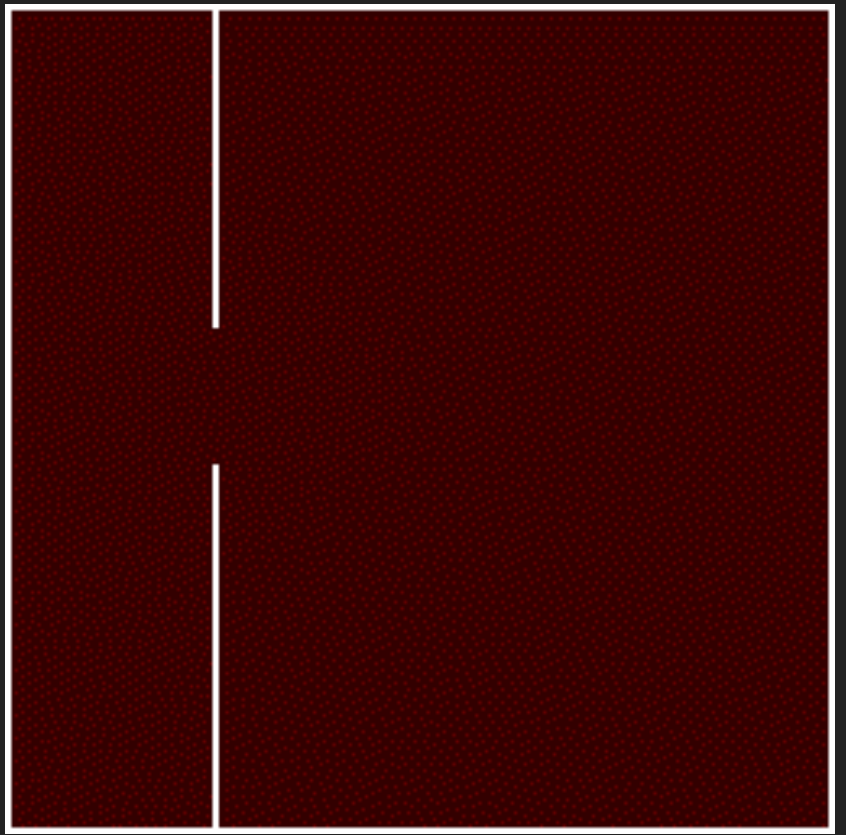
Stan początkowy:



Stan pośredni:



Stan końcowy – ustabilizowany:



Piotr Mirosław